# 3.2 Prefix Transposition Distance
# 3.3 Reversal Distance

Yi-Hean Lin

10/29/2012

# 3.2 Prefix Transposition Distance

We known that section 3.1 is transposition distance. Transposition is a method to solve the problem of sorting.

# 3.2 Prefix Transposition Distance

We known that section 3.1 is transposition distance. Transposition is a method to solve the problem of sorting.

**Definition ((3.1)Transposition)**

*For any permutation* $\pi$ *in* $S_n$, *the* *transposition* $\tau(i, j, k)$ *with* $1 \leq i < j < k \leq n+1$ *applied to* $\pi$ *exchanges the intervals determined respectively by* $i$ *and* $j-1$ *and by* $j$ *and* $k-1$, *transposition* $\pi$ *into* $\pi \circ \tau(i, j, k)$.

# 3.2 Prefix Transposition Distance

We known that section 3.1 is transposition distance. Transposition is a method to solve the problem of sorting.

**Definition ((3.1)Transposition)**

*For any permutation $\pi$ in $S_n$, the transposition $\tau(i, j, k)$ with $1 \leq i < j < k \leq n + 1$ applied to $\pi$ exchanges the intervals determined respectively by $i$ and $j - 1$ and by $j$ and $k - 1$, transposition $\pi$ into $\pi \circ \tau(i, j, k)$.*

Since the problem of sorting by transposition seems very difficult, some researchers have tried to study variants of this problem.

# 3.2 Prefix Transposition Distance

We known that section 3.1 is transposition distance. Transposition is a method to solve the problem of sorting.

**Definition ((3.1)Transposition)**

*For any permutation $\pi$ in $S_n$, the transposition $\tau(i, j, k)$ with $1 \leq i < j < k \leq n+1$ applied to $\pi$ exchanges the intervals determined respectively by $i$ and $j-1$ and by $j$ and $k-1$, transposition $\pi$ into $\pi \circ \tau(i, j, k)$.*

Since the problem of sorting by transposition seems very difficult, some researchers have tried to study variants of this problem.
In this section, we discuss a restricted version of sorting by transpositions in which only the "beginning" of the permutation may be moved.

### Definition (Prefix Transposition (Distance))

*A* *prefix transposition* *is a transposition* definition 3.1 $\tau(i, j, k)$ *with* $i = 1$.

.

## Definition (Prefix Transposition (Distance))

*A* *prefix transposition* *is a transposition* definition 3.1 $\tau(i, j, k)$ *with* $i = 1$*.* *The* *prefix transposition distance* *of a permutation* $\pi$ *will be denoted by* $ptd(\pi)$*.*

## Definition (Prefix Transposition (Distance))

*A* *prefix transposition* *is a transposition* definition 3.1 $\tau(i, j, k)$ *with* $i = 1$*.* *The* *prefix transposition distance* *of a permutation* $\pi$ *will be denoted by* $ptd(\pi)$*.*

Table 3.3 shows the distribution of the prefix transposition distance for $1 \leq n \leq 10$.

## Definition (Prefix Transposition (Distance))

*A* *prefix transposition* *is a transposition* definition 3.1 $\tau(i, j, k)$ *with* $i = 1$*. The* *prefix transposition distance* *of a permutation* $\pi$ *will be denoted by* $ptd(\pi)$*.*

Table 3.3 shows the distribution of the prefix transposition distance for $1 \leq n \leq 10$.

**Table 3.3**
The number of permutations $\pi$ in $S_n$ with $ptd(\pi) = k$; $1 \leq n \leq 10$

| $n \backslash k$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 1 | 6 | 14 | 3 | 0 | 0 | 0 | 0 | 0 |
| 5 | 1 | 10 | 50 | 55 | 4 | 0 | 0 | 0 | 0 |
| 6 | 1 | 15 | 130 | 375 | 194 | 5 | 0 | 0 | 0 |
| 7 | 1 | 21 | 280 | 1,575 | 2,598 | 562 | 3 | 0 | 0 |
| 8 | 1 | 28 | 532 | 4,970 | 18,096 | 15,532 | 1,161 | 0 | 0 |
| 9 | 1 | 36 | 924 | 12,978 | 85,128 | 188,386 | 74,183 | 1,244 | 0 |
| 10 | 1 | 45 | 1,500 | 29,610 | 308,988 | 1,364,710 | 1,679,189 | 244,430 | 327 |

# 3.2.1 Lower Bounds

Frist, we consider the lower bounds of $ptd(\pi)$.

# 3.2.1 Lower Bounds

Frist, we consider the lower bounds of $ptd(\pi)$.

**Lemma ([1])**

*For any permutation $\pi$, we have $ptd(\pi) \geq td(\pi)$.*

# 3.2.1 Lower Bounds

Frist, we consider the lower bounds of $ptd(\pi)$.

**Lemma ([1])**

*For any permutation $\pi$, we have $ptd(\pi) \geq td(\pi)$.*

**Proof.** It is a easy to see that every prefix transposition is a transposition,

# 3.2.1 Lower Bounds

Frist, we consider the lower bounds of $ptd(\pi)$.

**Lemma ([1])**

*For any permutation $\pi$, we have $ptd(\pi) \geq td(\pi)$.*

**Proof.** It is a easy to see that every prefix transposition is a transposition, we have the prefix transposition distance is always at least as large as the transposition distance ($ptd(\pi) \geq td(\pi)$);

# 3.2.1 Lower Bounds

Frist, we consider the lower bounds of $ptd(\pi)$.

**Lemma ([1])**

*For any permutation $\pi$, we have $ptd(\pi) \geq td(\pi)$.*

**Proof.** It is a easy to see that every prefix transposition is a transposition, we have the prefix transposition distance is always at least as large as the transposition distance ($ptd(\pi) \geq td(\pi)$); therefore, any lower bound on $td(\pi)$ is also a lower bound on $ptd(\pi)$. ∎

### Definition (Prefix Transposition Breakpoint)

*A* *prefix transposition breakpoint* *in a permutation* $\pi$ *in* $S_n$ *is a breakpoint of* $\pi$*, except that* $(0, \pi_1)$ *is always a prefix transposition breakpoint.*

### Definition (Prefix Transposition Breakpoint)

*A* *prefix transposition breakpoint* *in a permutation* $\pi$ *in* $S_n$ *is a breakpoint of* $\pi$*, except that* $(0, \pi_1)$ *is always a prefix transposition breakpoint.*
*The number of prefix transposition breakpoints in a permutation* $\pi$ *is denoted by* $ptd(\pi)$*.*

## Definition (Prefix Transposition Breakpoint)

*A* *prefix transposition breakpoint* *in a permutation* $\pi$ *in* $S_n$ *is a breakpoint of* $\pi$*, except that* $(0, \pi_1)$ *is always a prefix transposition breakpoint.* *The number of prefix transposition breakpoints in a permutation* $\pi$ *is denoted by* $ptd(\pi)$*.*

## Definition (k-Prefix Transposition)

*For a permutation* $\pi$*, a* *k-prefix transposition* *is a transposition* $\tau$ *such that* $ptb(\pi \circ \tau) = ptb(\pi) + k$*.*

The first important observation about prefix transposition breakpoints is the following.

page 8

The first important observation about prefix transposition breakpoints is the following.

**Lemma ([1]3.4)**

*For any $k$-prefix transposition $\tau$, we have $k \in \{-2, -1, 0, 1, 2\}$.*

page 8

The first important observation about prefix transposition breakpoints is the following.

**Lemma ([1]3.4)**

*For any $k$-prefix transposition $\tau$, we have $k \in \{-2, -1, 0, 1, 2\}$.*

**Lemma ([1]3.5)**

*For all $\pi$ in $S_n$, we have $ptd(\pi) \geq \lceil \frac{ptb(\pi)-1}{2} \rceil$.*

page 8

The first important observation about prefix transposition breakpoints is the following.

**Lemma ([1]3.4)**

*For any k-prefix transposition $\tau$, we have $k \in \{-2, -1, 0, 1, 2\}$.*

**Lemma ([1]3.5)**

*For all $\pi$ in $S_n$, we have $ptd(\pi) \geq \lceil \frac{ptb(\pi)-1}{2} \rceil$.*

**Proof.** Immediate from Lemma 3.4

page 8

## Lemma ([1])

*For all* $\pi$ *in* $S_n$*, there exist at most one prefix transportation* $\tau$ *such that* $ptb(\pi \circ \tau) = ptb(\pi) - 2$

;

## Lemma ([1])

*For all $\pi$ in $S_n$, there exist at most one prefix transportation $\tau$ such that $ptb(\pi \circ \tau) = ptb(\pi) - 2$*

## Lemma

*If a permutation $\pi$ is tight with respect to the lower bound of lemma 3.5, then $ptd(\pi) = td(\pi)$.*

;

## Lemma ([1])

*For all $\pi$ in $S_n$, there exist at most one prefix transportation $\tau$ such that $ptb(\pi \circ \tau) = ptb(\pi) - 2$*

## Lemma

*If a permutation $\pi$ is tight with respect to the lower bound of lemma 3.5, then $ptd(\pi) = td(\pi)$.*

Dias and Meidanis [1] show that a particular family of permutations can be sorted by prefix transpositions in linear time;

## Lemma ([1])

*For all $\pi$ in $S_n$, there exist at most one prefix transportation $\tau$ such that $ptb(\pi \circ \tau) = ptb(\pi) - 2$*

## Lemma

*If a permutation $\pi$ is tight with respect to the lower bound of lemma 3.5, then $ptd(\pi) = td(\pi)$.*

Dias and Meidanis [1] show that a particular family of permutations can be sorted by prefix transpositions in linear time; those are the permutations defined by
$(k+1\ k\ k+2\ k-1\ k+3\ k-2\ \cdots\ 2k-1\ 2\ 2k\ 1)$.

### Lemma ([1])

*For all $\pi$ in $S_n$, there exist at most one prefix transportation $\tau$ such that $ptb(\pi \circ \tau) = ptb(\pi) - 2$*

### Lemma

*If a permutation $\pi$ is tight with respect to the lower bound of lemma 3.5, then $ptd(\pi) = td(\pi)$.*

Dias and Meidanis [1] show that a particular family of permutations can be sorted by prefix transpositions in linear time; those are the permutations defined by
$(k+1\ k\ k+2\ k-1\ k+3\ k-2\ \cdots\ 2k-1\ 2\ 2k\ 1)$.
The prefix transposition distance of those permutations is $k$.

**Definition (clan)**

*A clan is a maximal interval of $\pi$ that contains only reverse adjacencies.*

## Definition (clan)

*A clan is a maximal interval of $\pi$ that contains only reverse adjacencies.*

Remark:If $\pi'_{i+1} = \pi'_i - 1$, it is called a reverse adjacency.

## Definition (clan)

*A clan is a maximal interval of $\pi$ that contains only reverse adjacencies.*

Remark:If $\pi'_{i+1} = \pi'_i - 1$, it is called a reverse adjacency.
A strip is a maximal interval of $\pi$ containing no breakpoint.

## Definition (clan)

*A clan is a maximal interval of $\pi$ that contains only reverse adjacencies.*

Remark:If $\pi'_{i+1} = \pi'_i - 1$, it is called a reverse adjacency.
A strip is a maximal interval of $\pi$ containing no breakpoint.

## Lemma ([2])

*For all $\pi$ in $S_n$, Let $\gamma(\pi)$ denote the set of all clans of $\pi$ of length at least 3, and $s(\pi)$ denote the number of strips of $\pi$. Then*

## Definition (clan)

*A* *clan* *is a maximal interval of* $\pi$ *that contains only reverse adjacencies.*

Remark:If $\pi'_{i+1} = \pi'_i - 1$, it is called a reverse adjacency.
A strip is a maximal interval of $\pi$ containing no breakpoint.

## Lemma ([2])

*For all* $\pi$ *in* $S_n$*, Let* $\gamma(\pi)$ *denote the set of all clans of* $\pi$ *of length at least 3, and* $s(\pi)$ *denote the number of strips of* $\pi$*. Then*

$$ptd(\pi) \geq \frac{s(\pi) + \frac{\sum_{C \in \gamma(\pi)} (|C| - 2)}{3}}{2}$$

Finally, Labarre[241] proved another lower bound and showed that its value is always at least as larger as that given by lemma 3.5.

definition 3.15

Finally, Labarre[241] proved another lower bound and showed that its value is always at least as larger as that given by lemma 3.5.

### Lemma (241)

*For all* $\pi$ *in* $S_n$*, we have*

definition 3.15

Finally, Labarre[241] proved another lower bound and showed that its value is always at least as larger as that given by lemma 3.5.

## Lemma (241)

*For all* $\pi$ *in* $S_n$*, we have*

$$ptd(\pi) \geq \frac{n+1+c(G(\pi))}{2} - c_1(G(\pi)) - \begin{cases} 0 & if\ \pi_1 = 1 \\ 1 & o.w \end{cases}$$

definition 3.15

Finally, Labarre[241] proved another lower bound and showed that its value is always at least as larger as that given by lemma 3.5.

## Lemma (241)

*For all* $\pi$ *in* $S_n$, *we have*

$$ptd(\pi) \geq \frac{n+1+c(G(\pi))}{2} - c_1(G(\pi)) - \begin{cases} 0 & \text{if } \pi_1 = 1 \\ 1 & o.w \end{cases}$$

*where* $c_1(G(\pi))$ *is the number of cycles of length* 1 *in* $G(\pi)$.
*and* $c(G(\pi))$ *is the number of alternating cycles in* $G(\pi)$

definition 3.15

Finally, Labarre[241] proved another lower bound and showed that its value is always at least as larger as that given by lemma 3.5.

## Lemma (241)

*For all $\pi$ in $S_n$, we have*

$$ptd(\pi) \geq \frac{n+1+c(G(\pi))}{2} - c_1(G(\pi)) - \begin{cases} 0 & \text{if } \pi_1 = 1 \\ 1 & o.w \end{cases}$$

*where $c_1(G(\pi))$ is the number of cycles of length 1 in $G(\pi)$. and $c(G(\pi))$ is the number of alternating cycles in $G(\pi)$*

## Definition (Cycle Graph $G(\pi)$)

*The cycle graph of a permutation $\pi$ of $\{1, 2, \ldots, n\}$ is the directed graph $G(\pi)$ with vertex set $\{0, 1, \ldots, n, n+1\}$ and whose arcs consist in*
*black (or reality) arcs $(\pi'_i, \pi'_{i-1})$ for $1 \leq i \leq n+1$,*
*gray (or desire) arcs $(i, i+1)$ for $0 \leq i \leq n$,*
*where $\pi'$ is the linear extension of $\pi$.*

definition 3.15

# 3.2.2 Upper Bounds

Now, we turn our attention to the upper bounds of $ptd(\pi)$.

# 3.2.2 Upper Bounds

Now, we turn our attention to the upper bounds of $ptd(\pi)$.
In the worst case, only one prefix transposition breakpoint can be removed at each step, and the fact that we can never remove $(0, \pi_1)$, yield the following upper bound.

# 3.2.2 Upper Bounds

Now, we turn our attention to the upper bounds of $ptd(\pi)$.
In the worst case, only one prefix transposition breakpoint can be removed at each step, and the fact that we can never remove $(0, \pi_1)$, yield the following upper bound.

**Lemma ([1])**

*For all $\pi$ in $S_n$, we have $ptd(\pi) \leq ptb(\pi) - 2$.*

## 3.2.2 Upper Bounds

Now, we turn our attention to the upper bounds of $ptd(\pi)$.
In the worst case, only one prefix transposition breakpoint can be removed at each step, and the fact that we can never remove $(0, \pi_1)$, yield the following upper bound.

**Lemma ([1])**

*For all $\pi$ in $S_n$, we have $ptd(\pi) \leq ptb(\pi) - 2$.*

Chitturi and Sudborough [2] proved another upper bound.

# 3.2.2 Upper Bounds

Now, we turn our attention to the upper bounds of $ptd(\pi)$.
In the worst case, only one prefix transposition breakpoint can be removed at each step, and the fact that we can never remove $(0, \pi_1)$, yield the following upper bound.

**Lemma ([1])**

*For all $\pi$ in $S_n$, we have $ptd(\pi) \leq ptb(\pi) - 2$.*

Chitturi and Sudborough [2] proved another upper bound.

**Theorem ([2])**

*For all $\pi$ in $S_n$, we have $ptd(\pi) \leq n - log_8\ n$.*

# 3.3 Reversal Distance

## Definition (Reversal)

*For any permutation* $\pi$ *in* $S_n$, *the* *reversal* $\rho(i,j)$ *with* $1 \leq i < j \leq n$ *applied to* $\pi$ *reverses the closed interval determined by* $i$ *and* $j$, *transforming* $\pi$ *into* $\pi \circ \rho(i,j)$.

# 3.3 Reversal Distance

## Definition (Reversal)

*For any permutation* $\pi$ *in* $S_n$*, the* *reversal* $\rho(i,j)$ *with* $1 \leq i < j \leq n$ *applied to* $\pi$ *reverses the closed interval determined by* $i$ *and* $j$*, transforming* $\pi$ *into* $\pi \circ \rho(i,j)$*. Therefore,* $\rho(i,j)$ *is the following permutation:*

$$\begin{pmatrix} 1 & \cdots & i-1 & \underline{i \; i+1 \; \cdots \; j-1 \; j} & j+1 & \cdots & n \\ 1 & \cdots & i-1 & j \; j-1 \; \cdots \; i+1 \; i & j+1 & \cdots & n \end{pmatrix}.$$

# 3.3 Reversal Distance

## Definition (Reversal)

*For any permutation* $\pi$ *in* $S_n$*, the* *reversal* $\rho(i,j)$ *with* $1 \leq i < j \leq n$ *applied to* $\pi$ *reverses the closed interval determined by* $i$ *and* $j$*, transforming* $\pi$ *into* $\pi \circ \rho(i,j)$*. Therefore,* $\rho(i,j)$ *is the following permutation:*

$$\begin{pmatrix} 1 & \cdots & i-1 & \underline{i \ i+1 \ \cdots \ j-1 \ j} & j+1 & \cdots & n \\ 1 & \cdots & i-1 & j \ j-1 \ \cdots \ i+1 \ i & j+1 & \cdots & n \end{pmatrix}.$$

## Definition (Reversal Distance)

*The* *reversal distance* *of a permutation* $\pi$ *will be denoted by* $rd(\pi)$*.*

# 3.3.1 Lower Bounds

Since a reversal can reduce the number of strong breakpoints by at most two (e.g., (1<u>32</u>4) $\rightarrow$ (1234)), we get the following first bound:

# 3.3.1 Lower Bounds

Since a reversal can reduce the number of strong breakpoints by at most two (e.g., (1<u>32</u>4) $\rightarrow$ (1234)), we get the following first bound:

**Theorem ([3])**

*For all $\pi$ in $S_n$, we have $rd(\pi) \geq \frac{sb(\pi)}{2}$.*

## Lemma ([4])

*Give permutation $\pi$, let $G = (V, E)$ be a graph*

## Lemma ([4])

*Give permutation* $\pi$, *let* $G = (V, E)$ *be a graph with the vertex set* $V$ *is the set of strong breakpoints of* $\pi$ $(V = \{(\pi'_i, \pi'_{i+1}) \mid |\pi'_{i+1} - \pi'_i| = 1\})$

## Lemma ([4])

*Give permutation* $\pi$, *let* $G = (V, E)$ *be a graph with the vertex set* $V$ *is the set of strong breakpoints of* $\pi$ $(V = \{(\pi'_i, \pi'_{i+1}) \mid \mid \pi'_{i+1} - \pi'_i \mid = 1\})$ *and whose edges connect two vertices such that the corresponding strong breakpoints can be removed using the signal reversal.*
$(E = \{\{(\pi'_i, \pi'_{i+1}), (\pi'_j, \pi'_{j+1})\} \mid \mid \pi'_i - \pi'_j \mid = 1 \text{ and } \mid \pi'_{i+1} - \pi'_{j+1} \mid = 1\})$

## Lemma ([4])

*Give permutation $\pi$, let $G = (V, E)$ be a graph with the vertex set $V$ is the set of strong breakpoints of $\pi$ $(V = \{(\pi'_i, \pi'_{i+1}) \mid \mid \pi'_{i+1} - \pi'_i \mid = 1\})$ and whose edges connect two vertices such that the corresponding strong breakpoints can be removed using the signal reversal.*
$(E = \{\{(\pi'_i, \pi'_{i+1}), (\pi'_j, \pi'_{j+1})\} \mid \mid \pi'_i - \pi'_j \mid = 1 \text{ and } \mid \pi'_{i+1} - \pi'_{j+1} \mid = 1\})$
*Let $m$ be the number of vertices in a maximum cardinality matching of $G$, then*

## Lemma ([4])

*Give permutation* $\pi$*, let* $G = (V, E)$ *be a graph with the vertex set* $V$ *is the set of strong breakpoints of* $\pi$ $(V = \{(\pi'_i, \pi'_{i+1}) | \mid \pi'_{i+1} - \pi'_i \mid = 1\})$ *and whose edges connect two vertices such that the corresponding strong breakpoints can be removed using the signal reversal.*
$(E = \{\{(\pi'_i, \pi'_{i+1}), (\pi'_j, \pi'_{j+1})\} | \mid \pi'_i - \pi'_j \mid = 1 \text{ and } \mid \pi'_{i+1} - \pi'_{j+1} \mid = 1\})$
*Let* $m$ *be the number of vertices in a maximum cardinality matching of* $G$*, then*
$rd(\pi) \geq \lceil \frac{m}{2} + \frac{2(sb(\pi) - m)}{3} \rceil$

## Lemma ([4])

*Give permutation* $\pi$, *let* $G = (V, E)$ *be a graph with the vertex set* $V$ *is the set of strong breakpoints of* $\pi$ $(V = \{(\pi'_i, \pi'_{i+1}) | \mid \pi'_{i+1} - \pi'_i \mid = 1\})$ *and whose edges connect two vertices such that the corresponding strong breakpoints can be removed using the signal reversal.*
$(E = \{\{(\pi'_i, \pi'_{i+1}), (\pi'_j, \pi'_{j+1})\} | \mid \pi'_i - \pi'_j \mid = 1 \text{ and } \mid \pi'_{i+1} - \pi'_{j+1} \mid = 1\})$
*Let* $m$ *be the number of vertices in a maximum cardinality matching of* $G$, *then*
$rd(\pi) \geq \lceil \frac{m}{2} + \frac{2(sb(\pi)-m)}{3} \rceil$

**Proof.** In the best case, two consecutive reversals can remove at most three strong breakpoints.

**Definition ((3.15)Breakpoint Graph)**

*The breakpoint graph of a permutation $\pi$ is the undirected graph $BG(\pi)$, whose vertex set is the vertex set of the cycle graph of $\pi$, and whose edges are the arcs of the cycle graph of $\pi$ taken without their orientation.*

## Definition ((3.15)Breakpoint Graph)

*The* *breakpoint graph* *of a permutation* $\pi$ *is the undirected graph* $BG(\pi)$*, whose vertex set is the vertex set of the cycle graph of* $\pi$*, and whose edges are the arcs of the cycle graph of* $\pi$ *taken without their orientation.*

Actually, breakpoint graph is the undirected version of the cycle graph $G(\pi)$ definition 3.3.

## Definition ((3.15)Breakpoint Graph)

*The breakpoint graph of a permutation $\pi$ is the undirected graph $BG(\pi)$, whose vertex set is the vertex set of the cycle graph of $\pi$, and whose edges are the arcs of the cycle graph of $\pi$ taken without their orientation.*

Actually, breakpoint graph is the undirected version of the cycle graph $G(\pi)$ definition 3.3.

Figure 3.4 shows an example of a breakpoint graph of a linear permutation.

## Definition ((3.15)Breakpoint Graph)

*The **breakpoint graph** of a permutation $\pi$ is the undirected graph $BG(\pi)$, whose vertex set is the vertex set of the cycle graph of $\pi$, and whose edges are the arcs of the cycle graph of $\pi$ taken without their orientation.*

Actually, breakpoint graph is the undirected version of the cycle graph $G(\pi)$ definition 3.3.

Figure 3.4 shows an example of a breakpoint graph of a linear permutation.

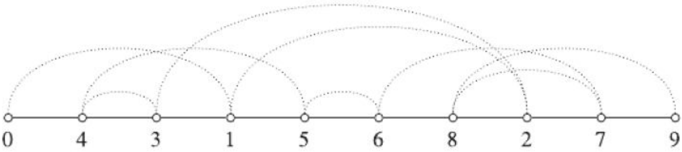


**Figure 3.4**
The breakpoint graph of the permutation (4 3 1 5 6 8 2 7)

Since each vertex has the same number of incident gray and black edges, the breakpoint graph decomposes into edge-disjoint alternating cycles.

Since each vertex has the same number of incident gray and black edges, the breakpoint graph decomposes into edge-disjoint alternating cycles.
It is easy to see, this decomposition is not unique

Since each vertex has the same number of incident gray and black edges, the breakpoint graph decomposes into edge-disjoint alternating cycles.
It is easy to see, this decomposition is not unique (this is actually the main reason why sorting by reversals is **NP**-hard; see Caprara [93]).

Since each vertex has the same number of incident gray and black edges, the breakpoint graph decomposes into edge-disjoint alternating cycles.
It is easy to see, this decomposition is not unique (this is actually the main reason why sorting by reversals is **NP**-hard; see Caprara [93]).
A decomposition in alternating cycles is maximal if it contain the largest number of alternating cycles.

Figure 3.5 is a maximal alternating cycle decomposition of the breakpoint graph of figure 3.4.

Figure 3.5 is a maximal alternating cycle decomposition of the breakpoint graph of figure 3.4.

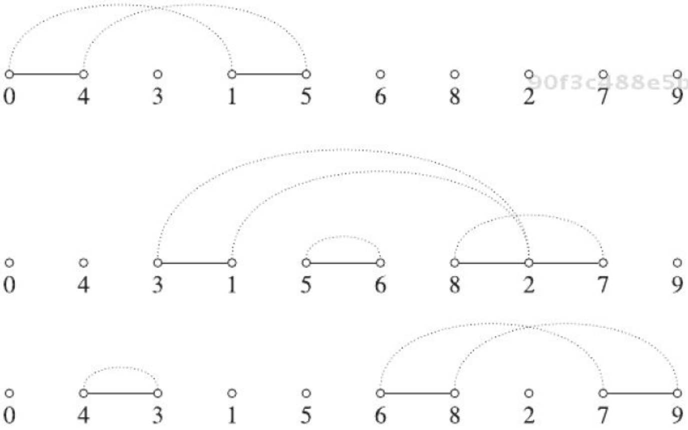




**Figure 3.5**
A maximal alternating cycle decomposition of the breakpoint graph of figure 3.4 into five cycles

We denote $c^*(BG(\pi))$ as the number of cycles in a maximal alternating cycle decomposition of $BG(\pi)$

We denote $c^*(BG(\pi))$ as the number of cycles in a maximal alternating cycle decomposition of $BG(\pi)$

The same kind of lower bound that was shown in the case of transpositions (theorems 3.3 and 3.4) is obtained for reversals,

We denote $c^*(BG(\pi))$ as the number of cycles in a maximal alternating cycle decomposition of $BG(\pi)$

The same kind of lower bound that was shown in the case of transpositions (theorems 3.3 and 3.4) is obtained for reversals,

### Theorem (3.3-4)

*For any permutation* $\pi$, *we have* $td(\pi) \geq \frac{p(\pi) - c(G(\pi))}{2}$.

*For any permutation* $\pi$, *we have* $td(\pi) \geq \frac{p(\pi) - c_{odd}(G(\pi))}{2}$.

We denote $c^*(BG(\pi))$ as the number of cycles in a maximal alternating cycle decomposition of $BG(\pi)$
The same kind of lower bound that was shown in the case of transpositions (theorems 3.3 and 3.4) is obtained for reversals,

### Theorem (3.3-4)

*For any permutation* $\pi$, *we have* $td(\pi) \geq \frac{p(\pi)-c(G(\pi))}{2}$.
*For any permutation* $\pi$, *we have* $td(\pi) \geq \frac{p(\pi)-c_{odd}(G(\pi))}{2}$.

### Theorem ([3])

*For any permutation* $\pi$, *we have* $rd(\pi) \geq \ p(\pi) - c^*(BG(\pi))$.

We denote $c^*(BG(\pi))$ as the number of cycles in a maximal alternating cycle decomposition of $BG(\pi)$

The same kind of lower bound that was shown in the case of transpositions (theorems 3.3 and 3.4) is obtained for reversals,

**Theorem (3.3-4)**

*For any permutation $\pi$, we have $td(\pi) \geq \frac{p(\pi)-c(G(\pi))}{2}$.*

*For any permutation $\pi$, we have $td(\pi) \geq \frac{p(\pi)-c_{odd}(G(\pi))}{2}$.*

**Theorem ([3])**

*For any permutation $\pi$, we have $rd(\pi) \geq p(\pi) - c^*(BG(\pi))$.*

But, the $c^*(BG(\pi))$ is not easy to compute.

## 3.3.2 Upper Bounds

We now turn our attention to the upper bounds of $rd(\pi)$.

## 3.3.2 Upper Bounds

We now turn our attention to the upper bounds of $rd(\pi)$.
The only upper bound on the unsigned reversal distance.

# 3.3.2 Upper Bounds

We now turn our attention to the upper bounds of $rd(\pi)$.
The only upper bound on the unsigned reversal distance.

**Lemma ([4])**

*For any permutation $\pi$, we have $rd(\pi) \leq sb(\pi) - 1$.*

# 3.3.2 Upper Bounds

We now turn our attention to the upper bounds of $rd(\pi)$.
The only upper bound on the unsigned reversal distance.

**Lemma ([4])**

*For any permutation $\pi$, we have $rd(\pi) \leq sb(\pi) - 1$.*

# 3.3.2 Upper Bounds

We now turn our attention to the upper bounds of $rd(\pi)$.
The only upper bound on the unsigned reversal distance.

**Lemma ([4])**

*For any permutation $\pi$, we have $rd(\pi) \leq sb(\pi) - 1$.*

It is trivial bound.

# 3.3.3.1 A Link with the Signed Version

There exists a version of the problem of sorting by reversals for signed permutations

# 3.3.3.1 A Link with the Signed Version

There exists a version of the problem of sorting by reversals for signed permutations, which will be discussed in detail in section 4.2.

# 3.3.3.1 A Link with the Signed Version

There exists a version of the problem of sorting by reversals for signed permutations, which will be discussed in detail in section 4.2.

**Definition (Spin)**

*A spin of a permutation $\pi$ in $S_n$ is a signed permutation $\overrightarrow{\pi}$ in $S_n^{\pm}$ such that $|\overrightarrow{\pi_i}| = \pi_i$ for all $1 \leq i \leq n$.*

# 3.3.3.1 A Link with the Signed Version

There exists a version of the problem of sorting by reversals for signed permutations, which will be discussed in detail in section 4.2.

**Definition (Spin)**

*A spin of a permutation $\pi$ in $S_n$ is a signed permutation $\overrightarrow{\pi}$ in $S_n^{\pm}$ such that $|\overrightarrow{\pi_i}| = \pi_i$ for all $1 \leq i \leq n$.*

**Lemma (198)**

*For any permutation $\pi$, in $S_n$, denote $\overrightarrow{\prod}$ as the set of all spins of $\pi$;*

# 3.3.3.1 A Link with the Signed Version

There exists a version of the problem of sorting by reversals for signed permutations, which will be discussed in detail in section 4.2.

## Definition (Spin)

*A spin of a permutation* $\pi$ *in* $S_n$ *is a signed permutation* $\overrightarrow{\pi}$ *in* $S_n^{\pm}$ *such that* $|\overrightarrow{\pi_i}| = \pi_i$ *for all* $1 \leq i \leq n$.

## Lemma (198)

*For any permutation* $\pi$*, in* $S_n$*, denote* $\overrightarrow{\prod}$ *as the set of all spins of* $\pi$*; we have ,*

$$rd(\pi) = min_{\overrightarrow{\pi} \in \overrightarrow{\prod}} srd(\overrightarrow{\pi}),$$

*where* $srd(\overrightarrow{\prod})$ *denotes the signed reversal distance of the spin* $\overrightarrow{\prod}$.

# 3.3.3.1 A Link with the Signed Version

There exists a version of the problem of sorting by reversals for signed permutations, which will be discussed in detail in section 4.2.

**Definition (Spin)**

*A spin of a permutation $\pi$ in $S_n$ is a signed permutation $\overrightarrow{\pi}$ in $S_n^{\pm}$ such that $|\overrightarrow{\pi_i}| = \pi_i$ for all $1 \leq i \leq n$.*

**Lemma (198)**

*For any permutation $\pi$, in $S_n$, denote $\overrightarrow{\prod}$ as the set of all spins of $\pi$; we have ,*

$$rd(\pi) = min_{\overrightarrow{\pi} \in \overrightarrow{\prod}} srd(\overrightarrow{\pi}),$$

*where $srd(\overrightarrow{\prod})$ denotes the signed reversal distance of the spin $\overrightarrow{\prod}$.*

A spin $\overrightarrow{\pi}$ of permutation $\pi$ is called optimal if $rd(\pi) = srd(\overrightarrow{\pi})$.

## Definition ((long) strong strip)

*A strong strip is a maximal interval containing no strong breakpoint*

### Definition ((long) strong strip)

*A strong strip is a maximal interval containing no strong breakpoint , and it is called long if its length is at least 3.*

## Definition ((long) strong strip)

*A* ***strong strip*** *is a maximal interval containing no strong breakpoint , and it is called* ***long*** *if its length is at least 3.*

## Lemma (198)

*For any permutation* $\pi$ *in* $s_n$*, there exists an optimal sorting sequence of reversals that never cuts long strips.*

### Definition ((long) strong strip)

*A strong strip is a maximal interval containing no strong breakpoint , and it is called long if its length is at least 3.*

### Lemma (198)

*For any permutation $\pi$ in $s_n$, there exists an optimal sorting sequence of reversals that never cuts long strips.*

### Lemma (198)

*For any permutation $\pi$ in $s_n$, there exists an optimal sorting sequence of reversals that never increases the number of strong breakpoints.*

- Z Dias, J Meidanis Sorting by prefix transpositions, A. H. F. Laender and A. L. Oliveria, eds., vol 2476 of Lecture Notes in Computer Science, Springer-Verlag, 2002, pp.65-76.
- B.Chitturi, H. Sudborough, W. Voit, and X. Feng, Bounding Prefix Transposition Distance for Strings and Permutations, IEEE Computer Society Press, 2008, p.468.
- V. Bafna, P. A. Pevzner, Genome rearrangements and sorting by reversals, IEEE Computer Society Press, 1993, p.148-157.
- J. Kececioglu and D. Sankoff, Exact and Approximation Algorithms for Sorting by Reversals, with Application to Genome Rearrangement, Algorithmica, 13(1995), pp.180-210.
- N, Tran, An easy case of sorting by reversals, A. Apostolico and J. Hein, eds., vol 1264 of Lecture Notes in Computer Science, Springer-Verlag, 1997, pp.83-89.